(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（A）**　(11)特許出願公開番号

特開平8－161509

(43)公開日　平成８年(1996)６月21日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｇ０６Ｔ　11/00					
15/00					
		9365－5H	Ｇ０６Ｆ　15/ 72	３５０	
		9365－5H		４５０　Ａ	

審査請求　有　　請求項の数９　ＯＬ　（全 15 頁）

(21)出願番号　特願平6－298618

(22)出願日　平成６年(1994)12月１日

(71)出願人　390009531
インターナショナル・ビジネス・マシーンズ・コーポレイション
INTERNATIONAL BUSINESS MASCHINES CORPORATION
アメリカ合衆国10504、ニューヨーク州　アーモンク　（番地なし）

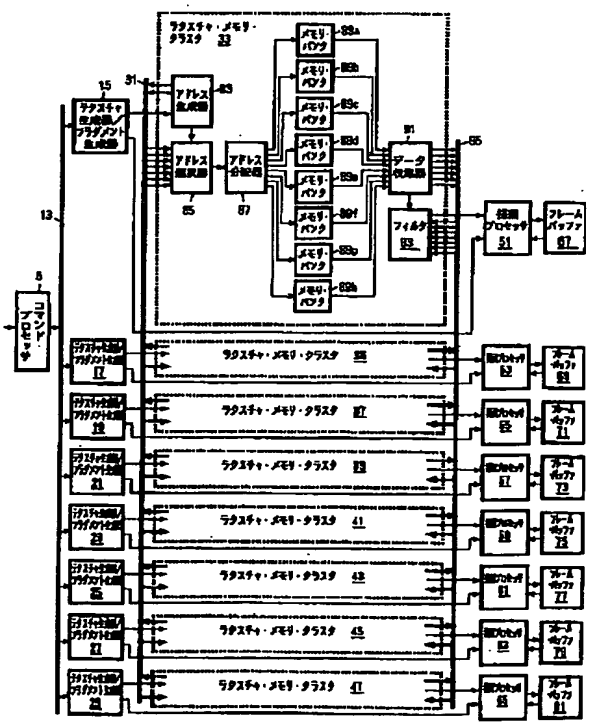
(72)発明者　川瀬　桂
神奈川県大和市下鶴間1623番地14　日本アイ・ビー・エム株式会社　東京基礎研究所内

(74)代理人　弁理士　合田　潔　（外２名）

最終頁に続く

(54)【発明の名称】　テクスチャ・マッピングを行うコンピュータ・システム

(57)【要約】
【目的】
【構成】複数のテクセルからなるテクスチャ・イメージ・データをインターリーブして重複がないように各々のメモリに格納する複数のテクスチャ・メモリ・クラスタであって、１のピクセルに対するテクスチャ・イメージの座標から、その座標に対応するテクスチャ値を計算するのに必要なテクセルを求め、そのテクセル値を複数のテクスチャ・メモリ・クラスタから収集する収集手段と、収集したテクセル値から１のピクセルに対するテクスチャ値を計算する手段とをそれぞれ有するものと、複数のメモリ・クラスタを接続するバスと、複数のメモリ・クラスタのうちの１つにそれぞれ接続され、１のピクセルに対するテクスチャ・イメージの座標を計算する複数のテクスチャ生成器とを有する。
【効果】メモリに重複するテクセルを保持することなく、効率的にテクセル値を各メモリ・クラスタ間で交換できる。



1

【特許請求の範囲】

【請求項１】複数のテクセルからなるテクスチャ・イメージ・データをインターリーブして重複がないように各々のメモリに格納する複数のテクスチャ・メモリ・クラスタであって、１のピクセルに対するテクスチャ・イメージの座標から、その座標に対応するテクスチャ値を計算するのに必要なテクセルを求め、そのテクセルの色情報であるテクセル値を前記複数のテクスチャ・メモリ・クラスタから収集する収集手段と、収集した前記テクセル値から前記１のピクセルに対するテクスチャ値を計算する手段とをそれぞれ有する前記複数のテクスチャ・メモリ・クラスタと、

前記複数のテクスチャ・メモリ・クラスタを接続するバスと、

前記複数のテクスチャ・メモリ・クラスタのうちの１つにそれぞれ接続され、前記１のピクセルに対するテクスチャ・イメージの座標を計算する複数のテクスチャ生成器と、

前記複数のテクスチャ・メモリ・クラスタのうちの１つにそれぞれ接続され、前記１のピクセルに対するテクスチャ値を用いて前記１のピクセルに表示すべきデータを生成する複数の描画プロセッサとを有するテクスチャ・マッピングを行うコンピュータ・システム。

【請求項２】前記収集手段が、

１のピクセルに対するテクスチャ・イメージの座標から、その座標に対応するテクスチャ値を計算するのに必要なテクセルを求め、前記複数のテクスチャ・メモリ・クラスタに通知する手段と、

前記メモリを検索し、通知された前記テクセルのテクセル値を得る検索手段と、

前記テクセル値を前記テクセルの通知元のテクスチャ・メモリ・クラスタに送信する手段と、

前記テクセル値を受信する手段とを含む請求項１記載のテスクチャ・マッピングを行うコンピュータ・システム。

【請求項３】前記メモリは複数のバンクに別れており、各バンクは、前記テクスチャ・メモリ・クラスタが格納するテクスチャ・イメージ・データをインターリーブして重複を生じないように格納することを特徴とする請求項１又は２記載のテスクチャ・マッピングを行うコンピュータ・システム。

【請求項４】前記複数のバンクは、入力ＦＩＦＯバッファと、出力ＦＩＦＯバッファとを含む請求項３記載のテスクチャ・マッピングを行うコンピュータ・システム。

【請求項５】前記検索手段は、

前記通知されたテクセルのアドレスを計算する手段と、計算された前記アドレスから該当する前記メモリのバンクに前記アドレスを分配する分配手段と、

前記メモリのバンクからテクセル値を得る手段とを含む請求項３又は４記載のテスクチャ・マッピングを行うコンピュータ・システム。

2

【請求項６】前記必要なテクセルの通知元のテクスチャ・メモリ・クラスタのＩＤを記憶するＦＩＦＯバッファを前記テクスチャ・メモリ・クラスタの各々がさらに有することを特徴とする請求項２乃至５のいずれか記載のテスクチャ・マッピングを行うコンピュータ・システム。

【請求項７】前記計算されたアドレスを分配した前記メモリのバンクのＩＤを記憶するＦＩＦＯバッファを前記テクスチャ・メモリ・クラスタの各々がさらに有することを特徴とする請求項５記載のテスクチャ・マッピングを行うコンピュータ・システム。

【請求項８】前記収集手段が、

前記１のピクセルに対するテクスチャ・イメージの座標を前記複数のテクスチャ・メモリ・クラスタに通知する手段と、

前記座標から、その座標に対応するテクスチャ値を計算するのに必要なテクセルであって、自己の前記メモリが保持するテクセルを計算する手段と、

前記テクセルのテクセル値を前記メモリから取り出す手段と、

前記テクセル値を前記テクセルの通知元のテクスチャ・メモリ・クラスタに送信する手段と、

前記テクセル値を受信する手段とを含む請求項１記載のテスクチャ・マッピングを行うコンピュータ・システム。

【請求項９】複数のテクセルからなるテクスチャ・イメージと、そのテクスチャ・イメージを１以上の段階で縮小したデータであり且つ１又は複数のテクセルからなる縮小テクスチャ・イメージとを含むテクスチャ・イメージ・データをインターリーブして重複を生じないよう各々のメモリに格納する複数のテクスチャ・メモリ・クラスタであって、１のピクセルに対する前記テクスチャ・イメージの座標及びピクセルと前記テクスチャ・イメージのテクセルの縮小率から前記１のピクセルに対するテクスチャ値を計算するのに必要なテクセルを求め、そのテクセルの色情報であるテクセルの値を前記複数のテクスチャ・メモリ・クラスタが出力するように要求する手段と、要求された前記テクセルの値を検索し、要求した前記テクスチャ・メモリ・クラスタに出力する手段と、前記テクセルの値を収集する手段と、収集された前記テクセルの値から前記テクスチャ値を計算する手段とをそれぞれ有する前記複数のテクスチャ・メモリ・クラスタと、

前記複数のテスクチャ・メモリ・クラスタを接続するバスと、

前記複数のテクスチャ・メモリ・クラスタのうちの１つにそれぞれ接続され、前記１のピクセルに対するテクスチャ・イメージの座標及び前記縮小率を計算する複数のテクスチャ生成器と、

前記複数のテクスチャ・メモリ・クラスタのうちの１つにそれぞれ接続され、前記１のピクセルに対するテクスチャ値を用いて前記１のピクセルに表示すべきデータを生成する複数の描画プロセッサとを有するテクスチャ・マッピングを行うコンピュータ・システム。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、コンピュータ・グラフィクス装置にかかり、特に、コンピュータを用いて所定の領域へ模様等を張り付けるテクスチャ・マッピングを用いたコンピュータ・グラフィクス装置に関する。

【０００２】

【従来の技術】コンピュータ・グラフィクスでは、様々な模様、すなわちイメージ（Texture,テクスチャ）を張り付けるテクスチャ・マッピング(Texture Mapping）によって画像を形成することがある。テクスチャ・マッピング(Texture Mapping）とは、滑らかな平面や曲面上に、木目やスイカの表面の縞模様等の厚みのないパターンやイメージ(Texture）を張ること(Mapping、以下、マッピングという。）である。例えば、フライト・シミュレータ（米国マイクロソフト社の商標）は、予め撮影しておいた景色の写真イメージを背景部分にテクスチャ・マッピングすることで、仮想現実的な映像を高速で提供している。

【０００３】しかしながら、このテクスチャ・マッピングは大量のデータ・アクセス及び演算量を必要とするので、実時間処理するためには、広いバンド幅を有するテクスチャ・メモリ及び専用のハードウェアを必要としていた。このテクスチャ・メモリは、張り付けるべきパターンや背景等のイメージを表す２次元の配列データ（以下、テクスチャ・イメージ・データという。）を記憶するためのメモリである。このテクスチャ・メモリは、高速なアクセスを必要とすると共に大容量を必要とするために、コスト高であった。

【０００４】近年のコンピュータの大容量化、小型化及び高速化の加速に伴って、コンピュータ・グラフィクスを利用するコンピュータ・システムでは、高速かつ高効率にテクスチャ・マッピングが可能なハードウェアを低価格で提供することが要求されると共に、課題となっている。以下、テクスチャ・マッピングについて詳述する。

【０００５】[テクスチャ・マッピングにおける座標系]コンピュータ・グラフィクスでは、次の３種類の座標系を用いて物体を表現することが一般的である。

【０００６】第１の座標系は、モデル座標系（Model coordinate system）である。このモデル座標系は、個々の物体に固有な座標系、すなわち、物体を記述するのに適宜選択される座標系である。図５（Ａ）には、一例として、Ｘ軸、Ｙ軸、Ｚ軸の各々が直交する直交座標系における、物体としての立方体ＯＢを示した。

【０００７】第２の座標系は、視点座標系（Eye coordinate system）である。この視点座標系は、物体を実質的に目視した時の目の位置に合致された座標系、すなわち、図５（Ｃ）に示すように、視線と１つの座標軸とが合致する座標系である。

【０００８】第３の座標系は、スクリーン座標系（Screen coordinate system）である。このスクリーン座標系は、図５（Ｄ）に示すように、物体を含む３次元のシーンが投影された２次元のスクリーンＳＣ（図５（Ｃ）参照）における座標系である。図５（Ｃ）おけるスクリーン座標系は、直交する横軸のＸ＿Ｓ軸、及び縦軸のＹ＿Ｓ軸によって構成されている。

【０００９】テクスチャ・マッピングにおいては、上記の３種類の座標系の他に、図５（Ｂ）に示すように、テクスチャ・イメージ・データを記述するための座標系（以下、テクスチャ座標系：Texture coordinate system、という。）が必要である。図５（Ｂ）におけるテクスチャ座標系は、直交する横軸のＵ軸、及び縦軸のＶ軸によって構成されている。テクスチャ・マッピングの実質的な動作は、スクリーン座標系で本来の物体の色を描画すべき点に、テクスチャ座標で指定されるテクスチャ・イメージ・データの色を描画することである。

【００１０】上記のモデル座標系は、複数の基本的物体を組み合わせた複雑な物体を扱うときに、便宜上、各基本的物体を簡単に記述するための座標系である。従って、複雑な物体に対してテクスチャ・マッピングをしてスクリーン座標系で記述するためには、これら複数の基本的物体を纏めた複雑な物体を視点座標系に直す。ここでは、立方体ＯＢに対して、上面にシェーディング（Shading）をすると共に、側面にテクスチャ・マッピングする場合を説明する（図５（Ｃ）の立方体ＯＢ1参照）。この後に、記述された複数の基本的物体をスクリーンＳＣに投影することを想定する。この時点では、複雑な物体（立方体ＯＢ１）はスクリーン座標系で記述される。なお、以下の説明を簡単にするため、取り扱う物体は、既にスクリーン座標系に変換されているものとする。

【００１１】[テクスチャ・マッピングの詳細]図６（Ａ）に示すように、テクスチャ・イメージ・データは、平面上に単位正方形（１辺の長さが１の正方形）が縦横に配置され、所定の大きさの正方形（図６（Ａ）では、８×８の単位正方形で形成される正方形領域）を形成している。この格子点、すなわち、ｉ番目の列（ｉ：０≦ｉ≦７）でｊ番目の行（ｊ：０≦ｊ≦７）に対応する格子点（図中の×マーク）を中心とする各々の単位正方形の中心には、張り付けるべきイメージの色情報が格納されている。

【００１２】以下、これらの各単位正方形をテクセル、単位正方形の中心をテクセル中心、テクセル中心に格納されている色情報をテクセル値、及びテクセルが配置さ

れた所定の大きさの正方形領域をテクスチャという。また、テクスチャの一辺に含まれるテクセルの個数（図６（Ａ）では、８個）をテクスチャのサイズという。以下の説明を簡略にするため、ｊ番目の行でかつｉ番目の列のテクセル（ｉ，ｊ）のテクセル値をＴ［ｉ，ｊ］と略記する。

【００１３】次に、テクスチャを曲面に張り付ける場合、通常の曲面はポリゴン分割されていており、視点座標系ではテクスチャ・マッピングとはポリゴンとテクスチャの間の線形変換になる。これによって、ポリゴン上の各点とテクスチャ内の各点は厳密に１対１で対応し、さらに、この対応はテクスチャをポリゴンに張り付ける時点で完全に決定できる。最終的には、画面上（スクリーン上）に投影されたポリゴン内の各ピクセルの色を決定しなければならない。従って、スクリーン座標系は視点座標系の各ピクセルとテクスチャ内の各テクセルの間で対応しなければならないが、スクリーン座標系は視点座標系をスクリーンＳＣ上に投影したものであるので、ピクセル（ｘ＿ｓ，ｙ＿ｓ）に対応するテクスチャ内の点Ｐ（ｕ，ｖ）の座標は定まっている。この対応を用いてピクセル（ｘ＿ｓ，ｙ＿ｓ）の色を決定する。一般に用いられている色決定のアルゴリズムは、オリジナルのテクスチャ・イメージ・データと以下に説明するミップマップ（Mipmap）と呼ばれるオリジナル・データをさらに加工したデータを用いる方法、及びミップマップを用いずにオリジナルのテクスチャ・イメージ・データ等を用いる方法に大別される。

【００１４】ミップマップを用いない方法には２つの方法があり、その１つは点Ｐを含むテクセルのテクセル値をピクセルの色とする方法である。すなわち、点Ｐに最も近いテクスチャ内のテクセル（ｉ。，ｊ。）のテクセル値Ｔ［ｉ。，ｊ。］をピクセルの色とする。引数ｉ。，ｊ。は各々ｕ＋０．５，ｖ＋０．５の整数部として求めることができる。これは３次元グラフィクス・インタフェイスのＡＰＩ（Application Program Interface）の１種であるOpenGL(Mar Segal, Akeley, The OpenGL Graphics System:A Specification Version 1.0)では、テクセル・ニアレスト（Texel Nearest）と呼ばれる方法である。

【００１５】しかしながら、テクスチャ・マッピングされるポリゴンが近くにあり、スクリーン座標系上では非常に拡大される場合、１つのテクセル値が多数のピクセルに対応することになるので、タイルをはったようになり、ギザギザを生ずる。

【００１６】他の方法は、上記を改良した第２のテクセル・リニア・インタポーレイテッド（Texel Linear Interpolated、略して、Texel Linearともいう。以下、テクセルリニアという）である。この場合、図６（Ａ）に示すように、点Ｐを囲む４つのテクセルの値を線形補間するというものである。すなわち、点Ｐ（ｕ，ｖ）を囲むテクセル中心を（ｉ。，ｊ。），（ｉ。＋１，ｊ。），（ｉ。，ｊ。＋１），（ｉ。＋１，ｊ。＋１）（図中の太線の×マーク）としてａ＝ｕの少数部分、ｂ＝ｖの少数部分とおいたときに、以下の式で得られる値をピクセルの色とするものである。

$$(1-a)\times(1-b)\times T[i_0, j_0] + (1-a)\times b\times T[i_0, j_0+1] + a\times(1-b)\times T[i_0+1, j_0] + a\times b\times T[i_0+1, j_0+1]$$

引数ｉ。，ｊ。はｕ，ｖの整数部分として求めることができる。他の方法は上式から理解されるように線形補間を２回続けて行った場合と等価であるのでバイリニアともいう。

【００１７】次に、ミップマップを用いる方法を説明する。上記の方法では、投影されたポリゴン上の各点におけるテクセルの縮小拡大率がピクセルの位置によって大きく変化するので、スクリーンとのなす角度が大きなポリゴンが投影された時に不都合を生ずる。すなわち、スクリーン上極端に縮小されるテクセルに対しては上記のように４テクセルの重みつき平均では足りず、多くのテクセルが必要である。この必要とするテクセルは、ピクセルーテクセル間の縮小拡大率（Level Of Detail, ピクセルとテクセルの対応関係から演算できる。線型補間計算を行う時に必ず生ずる。以下、ＬＯＤという。）から求めることができる。このようにピクセル毎に平均のとり方を変化させる方法（コンボリューションと呼ばれる方法）は効果的だが、効率的でなく、実装が困難であるので以下に説明するミップマップを用いて代用するのが一般的である。

※【００１８】ミップマップは、先ずオリジナルのテクスチャ中のテクセルを、図６（Ａ）の太線で囲まれる縦横２個ずつのテクセルをブロックに纏める。次に、ブロック内に含まれる４つのテクセルのテクセル値を平均することによってブロックのテクセル値を求める。次に、図６（Ｂ）に示すように、求めたテクセル値に対応させるための新しい１つのテクセルを生成すると共に、元のテクスチャにおけるブロックの順序に配列し、全てのブロックについて行うことによって新しいテクスチャを生成する。このブロックで纏めることにより生成された新しいテクスチャと、生成前のオリジナル・テクスチャとを区別するために、オリジナル・テクスチャ（図６（Ａ））をレベル０のミップマップ、そしてここではサイズが半分になった新しいテクスチャ（図６（Ｂ））をレベル１のミップマップという。次に、レベル１のミップマップを同様に処理してレベル２のミップマップを生成する。この生成されたテクスチャが、サイズ１のテクスチャになるまで上記と同様の処理を繰り返し、高レベルのミップマップを生成する。なお、このときのテクセル値がミップマップのレベルに依存することは勿論であ

る。これを表すため上記の記号を拡張してレベルｋのミップマップ中のテクセル（ｉ，ｊ）のテクセル値をＴ［ｉ，ｊ：ｋ］と表記する。

【００１９】このミップマップを使ってテクスチャ・マッピングするのに先だって、上記のＬＯＤから用いるべきミップマップのレベルを決定しなければならない。OpenGLでは細かい規定がなされているが、ここでは必要な部分のみに簡単化して説明する。上記のテクセル・ニアレスト、テクセル・リニアと同様にミップマップ・ニアレスト、ミップマップ・リニアの概念があり、合計４種類のテクセル・ミップマップ法がある。なお、各レベルのミップマップ中のピクセルに対応する点の座標Ｐ（ｕ，ｖ）はレベルｋに依存したＰ（u_k，v_k）となるので、注意しなければならない。

【００２０】次に、テクセル・ミップマップ法の４種類１から４の各々を説明する。

1. ニアレストーミップマップーニアレスト（Nearest-Mipmap-Nearest）

ｋ＝（ＬＯＤ＋０．５の整数部分）としてレベルｋのミップマップ中にピクセルに対応する点P_k（ｕ，ｖ）を求め、点P_kを含むテクセル値Ｔ［i_o，j_o：ｋ］をピクセルの色とする。

2. リニアーミップマップーニアレスト（Liner-Mipmap-Nearest）

ｋ＝（ＬＯＤ＋０．５の整数部分）としてレベルｋのミップマップ中にピクセルに対応する点P_k（u_k，v_k）を求め、点P_kを囲む４つのテクセルのテクセル値を線形補間する。これは「レベルｋ」ということ以外は上記で説明したテクセルリニアと同様である。

3. ニアレストーミップマップーリニア（Nearest-Mipmap-Liner）

ｋ＝（ＬＯＤの整数部分）としてレベルｋとレベルｋ＋１の各々のミップマップ中のピクセルに対応する点：P_k，P_{k+1}を各々含むテクセルのテクセル値をｄ＝（ＬＯＤの少数部分）を内分比として線形補間する。

4. リニアーミップマップーリニア（Liner-Mipmap-Liner）

ｋ＝（ＬＯＤの整数部分）とする。レベルｋのミップマップ中のピクセルに対応する点P_kを含む４つのテクセルのテクセル値を線形補間した（テクセルリニア）結果と、レベルｋ＋１のミップマップ中のピクセルに対応する点P_{k+1}を含む４つのテクセルのテクセル値を線形補間した（テクセルリニア）結果を、ｄ＝（ＬＯＤの少数部分）を内分比として線形補間（ミップマップリニア）する。

【００２１】以上説明したようなミップマップ法を用いて、高速にテクスチャ・マッピングを行うためのコンピュータ・グラフィクス装置には、次の点が要求される。

・テクスチャ・メモリの容量が大きいこと

・テクスチャ・メモリとして用いるメモリのアクセス速度が高速なこと

・テクスチャ・メモリを効率的に利用すること

【００２２】容量の点を考えると、テクスチャ・メモリはテクスチャのテクスチャ・イメージ・データを格納する関係上、数Ｍバイトの容量が必要である。このため、テクスチャ・メモリには一般的に低コストのＤＲＡＭが用いられる。

【００２３】そして速度の点を考えると、このＤＲＡＭのサイクルタイムは８０ｎｓ程度なので、１２．５Ｍテクセル／秒程度で動作させることができる。但し、上記で述べたようにリニアーミップマップーリニア方式では同時に８つのテクセル値を参照しなければならない。８つのテクセル値を同時に参照するには、ミップマップのレベル毎にインターリーブをかけ、さらに各々のレベルをｉ，ｊごとにインターリーブして８つのメモリに格納してリニアーミップマップーリニアに必要なテクセル値を同時に参照できるような構成にしなければならない。そうしないと、１００Ｍテクセル／秒程度のアクセス速度を出すことができない。

【００２４】また、線形補間の計算をする回路（補間回路、Interpolator）が８つのメモリチップに直結している必要がある。各メモリチップに対するアドレスバスと各メモリチップから補間回路へのデータバスのビット幅は、アドレスが２０ビット、テクスチャ・イメージ・データが１６ビットとしても、（２０＋１６）×８＝２８８ビットとなり、配線が複雑化すると共に、チップのピンネック（１チップから出すことのできる信号線の制限）という問題も発生する。

【００２５】[従来例]上記の要求を満たす一例として、コンピュータ・グラフィクス装置に対応するテクスチャ・マッピング処理方法がある（特開昭６２－１４０１８１号公報参照）。このテクスチャ・マッピング処理方法では、プロセッサ素子内部にテクスチャ・イメージ・データを分割して有し、プロセッサ間通信路を介して他のプロセッサ素子内部のテクスチャ・イメージ・データを受けとっている。

【００２６】しかしながら、このコンピュータ・グラフィクス装置では、次のような問題がある。

・基本的にソフトウエアによって処理するための機構であり、動作が複雑でハードウエア化が困難であり、高速動作が期待できない。

・描画を処理する部分とテクスチャを処理する部分が１対１に対応しているので、各々の比率を変化できない。このため、描画プリミティブの種類や大きさ、それに貼付けるテクスチャイメージの大きさに対する、パフォーマンス要求に対して、テクスチャを処理する部分の量を変化させることができない。

・他のプロセッサ素子に対してプロセッサ間通信路を介して、表示すべき画素の情報を送り、その送信した情報に基づく演算結果を受けとるというプロトコルなので、

プロトコルのオーバーヘッドが大きい。また、ミップマップを用いた方法を直接適用しようとすると、テクセルを全く重複しないように保持することができないので、各プロセッサ内のメモリの容量が大きくなければならない。

【００２７】また、他例であるコンピュータ・グラフィクス装置として、リアリティ・エンジン・グラフィクス(Reality Engine Graphics、COMPUTER GRAPHICS Proceedings, Annual Conference Series, 1993pp-116 参照）がある。

【００２８】図７に示すように、このコンピュータ・グラフィクス装置９０は、コマンド・プロセッサ１２を備えている。コマンド・プロセッサ１２は、コマンド・バス１６を介してフラグメント生成器(Fragment Generator)７２、７４、７６、７８に接続されている。

【００２９】フラグメント生成器７２の出力側の一方はテクスチャ・メモリとしてのテクスチャ・メモリ・クラスタ(TEXTURE MEMORY CLUSTER)２２を介して描画プロセッサ(Drawing Processor) ４２に接続されており、出力側の他方は描画プロセッサ４２に直接接続されている。描画プロセッサ４２はフレーム・バッファ５２に接続されている。このフレーム・バッファ５２の出力側は、ディスプレイ６０に接続されている。テクスチャ・メモリ・クラスタ２２は、アドレス生成器６２、メモリ（メモリ・バンク）６４Ａ，６４Ｂ，６４Ｃ，６４Ｄ，６４Ｅ，６４Ｆ，６４Ｇ，６４Ｈ及びフィルタ６６から構成されており、アドレス生成器６２は各メモリ６４Ａ～６４Ｈに接続され、各メモリ６４Ａ～６４Ｈはフィルタ６６に接続されている。

【００３０】また、フラグメント生成器７４の出力側の一方はテクスチャ・メモリ・クラスタ２４を介してフレーム・バッファ５４に接続された描画プロセッサ４４に接続されており、他方は描画プロセッサ４４に直接接続されている。同様に、フラグメント生成器７６の一方はテクスチャ・メモリ・クラスタ２６を介してフレーム・バッファ５６に接続された描画プロセッサ４６に接続されており、他方は描画プロセッサ４６に直接接続されている。フラグメント生成器７８の出力側の一方はテクスチャ・メモリ・クラスタ２８を介してフレーム・バッファ５８に接続された描画プロセッサ４８に接続されており、他方は描画プロセッサ４８に直接接続されている。また、上記のフレーム・バッファ５４、５６、５８の出力側は、ディスプレイ６０に接続されている。

【００３１】なお、テクスチャ・メモリ・クラスタ２４、２６、２８は、テクスチャ・メモリ・クラスタ２２と同様の構成のため、詳細な説明を省略する。

【００３２】また、このコンピュータ・グラフィクス装置９０では、スクリーンについて縦方向を４列に分割して、分割された各々に属するピクセルをテクスチャ・メモリ・クラスタ２２、２４、２６、２８が担当し処理している。

【００３３】次に、コンピュータ・グラフィクス装置９０の動作を説明する。先ず、コマンド・プロセッサ１２は、その下流にあるフラグメント生成器７２～７８、テクスチャ・メモリ・クラスタ２２～２８、及び描画プロセッサ４２～４８の各プロセッサに対する命令を解釈すると共に、その命令等をコマンドバス１６に出力する。この命令には、各々のプロセッサ４２～４８中の図示しないレジスタにデータを格納するコントロール・コマンドや描画コマンドがある。スクリーン上における描画は、三角形単位(ポリゴン単位)で行われる。コマンド・プロセッサ１２は、この描画コマンドを受け取ると、それに続く三角形の頂点単位のデータ（三角形の頂点座標等）を受け取る。このデータのフォーマットを以下の図８に示す。

但し、ｘ＿ｓ，ｙ＿ｓ：三角形の頂点のスクリーン座標系での座標

Ｒ，Ｇ，Ｂ，Ａ：スクリーンＳＣ上の座標（ｘ＿ｓ，ｙ＿ｓ）における色情報

ｕ，ｖ　　：座標（ｘ＿ｓ，ｙ＿ｓ）にテクスチャ・マッピングすべきテクスチャ・イメージ・データのテクスチャ座標系における座標

【００３４】これらのデータは、フラグメント生成器７２、７４、７６、７８の各々に、コマンドバス１６を介して同時にブロードキャストされる。各フラグメント生成器は、得られたデータを補間して三角形内部のピクセル単位のデータを生成する。このとき生成されたピクセルのデータが、フラグメント生成器自身が担当するピクセルのデータであれば、ピクセル単位に補間されたデータ（以下の図９に示す）を、対応する描画プロセッサ４２～４８に出力する。

【００３５】これと共に、フラグメント生成器は，以下の図１０に示すデータ、すなわち、三角形の頂点のスクリーン座標系での座標、及び座標（ｘ＿ｓ，ｙ＿ｓ）にテクスチャ・マッピングすべきテクスチャ・イメージ・データのテクスチャ座標系における座標（ｕ，ｖ）を、対応するテクスチャ・メモリ・クラスタに出力する。

【００３６】このとき、フラグメント生成器自身が担当するピクセルでない場合は、描画プロセッサにおいてデータは破棄されるが、ピクセルは必ず何れかのフラグメント生成器が担当しているので、全く処理されないピクセルが生じることはない。テクスチャ・メモリ・クラスタのアドレス生成器は、この座標（ｕ，ｖ）に基づいて、上述した例えばリニアーミップマップーリニアの計算に必要なテクセルを求め、そのテクセルのメモリにおける物理アドレスを計算し、出力する。この物理アドレスをもとにメモリはテクセル値をフィルタに出力し、フィルタは出力されたテクセル値をもとに例えばリニアーミップマップーリニアに必要な計算を行って、ピクセルに張り付けるべき、テクスチャ・メモリ・クラスタから

の色情報を、対応する描画プロセッサに出力する。描画プロセッサでは、予め入力されたピクセルの色情報とテクスチャ・メモリ・クラスタからの色情報を考慮して、対応するフレーム・バッファに格納されている該当ピクセルの値に反映させる。

【００３７】なお、各テクスチャ・メモリ・クラスタ２２～２８には、予め全て同一の内容のテクスチャ・イメージ・データが重複して格納されている。従って、４つのテクスチャ・メモリ・クラスタ２２～２８による並列効果によって性能向上、特に処理速度の高速化を図ることができが、重複してデータを保持すれば高価なメモリの容量は増大する。

【００３８】

【発明が解決しようとする課題】本発明は、上記事実を考慮して、コンピュータ・グラフィクス・インタフェイスにあるテクスチャ・マッピングの機能を効率良く実行させることができるコンピュータ・グラフィクス装置を得ることが目的である。

【００３９】特に、必要とするメモリの容量を劇的に減少させても効率が許容範囲内の悪化で済むシステムを得ることが目的である。

【００４０】

【課題を解決するための手段】上記目的を達成する本発明は、複数のテクセルからなるテクスチャ・イメージ・データをインターリーブして重複がないように各々のメモリに格納する複数のテクスチャ・メモリ・クラスタであって、ある１つのピクセルに対するテクスチャ・イメージの座標から、その座標に対応するテクスチャ値を計算するのに必要なテクセルを求め、そのテクセルの色情報であるテクセル値を複数のテクスチャ・メモリ・クラスタから収集する収集手段と、収集したテクセル値から１のピクセルに対するテクスチャ値を計算する手段とをそれぞれ有する複数のテクスチャ・メモリ・クラスタと、複数のテクスチャ・メモリ・クラスタを接続するバスと、複数のテクスチャ・メモリ・クラスタのうちの１つにそれぞれ接続され、前述の１のピクセルに対するテクスチャ・イメージの座標を計算する複数のテクスチャ生成器と、複数のテクスチャ・メモリ・クラスタのうちの１つにそれぞれ接続され、前述の１のピクセルに対するテクスチャ値を用いて前述の１のピクセルに表示すべきデータを生成する複数の描画プロセッサとを有するテクスチャ・マッピングを行うコンピュータ・システムである。このようにすると、各テクスチャ・メモリ・クラスタ内のメモリに重複するテクセルを保持することなく、効率的にテクセル値を各テクスチャ・メモリ・クラスタ間で交換することができる。

【００４１】また、上述の収集手段は、ある１つのピクセルに対するテクスチャ・イメージの座標から、その座標に対応するテクスチャ値を計算するのに必要なテクセルを求め、複数のテクスチャ・メモリ・クラスタに通知する手段と、メモリを検索し、通知されたテクセルのテクセル値を得る検索手段と、テクセル値をテクセルの通知元のテクスチャ・メモリ・クラスタに送信する手段と、テクセル値を受信する手段とを含むようにすることも考えられる。

【００４２】さらに、上述のメモリは複数のバンクに別れており、各バンクは、テクスチャ・メモリ・クラスタが格納するテクスチャ・イメージ・データをインターリーブして重複を生じないように格納するようにすることも考えられる。これにより、同一テクスチャ・メモリ・クラスタ内のメモリに同一テクセル値を重複して保持する、ということを完全に排除することができる。

【００４３】また、先の複数のバンクは、入力ＦＩＦＯバッファと、出力ＦＩＦＯバッファとを含むことも考えられる。このようにすると、テクセルをそのアドレスでインターリーブして格納した場合に、同一バンクに対するアクセスの集中によるパフォーマンス低下を抑えることができる。

【００４４】また、上述の検索手段は、通知されたテクセルのアドレスを計算する手段と、計算されたアドレスから該当するメモリのバンクにアドレスを分配する分配手段と、メモリのバンクからテクセル値を得る手段とを含むことも考えられる。このように分配することにより、効率よくメモリにアクセスすることができる。

【００４５】さらに、先の必要なテクセルの通知元のテクスチャ・メモリ・クラスタのＩＤを記憶するＦＩＦＯバッファをテクスチャ・メモリ・クラスタの各々がさらに有するようすることも考えられる。このようにすると、必要なテクセルのテクセル値を必要なテクスチャ・メモリ・クラスタに送信することができる。

【００４６】また、先の計算されたアドレスを分配した前記メモリのバンクのＩＤを記憶するＦＩＦＯバッファをテクスチャ・メモリ・クラスタの各々がさらに有するようにすることも考えられる。このようにすると、検索結果を適切なメモリ・バンクから出力させることができる。

【００４７】さらに、上述の収集手段が、ある１つのピクセルに対するテクスチャ・イメージの座標を複数のテクスチャ・メモリ・クラスタに通知する手段と、座標から、その座標に対応するテクスチャ値を計算するのに必要なテクセルであって、自己のメモリが保持するテクセルを計算する手段と、テクセルのテクセル値をメモリから取り出す手段と、テクセル値をテクセルの通知元のテクスチャ・メモリ・クラスタに送信する手段と、テクセル値を受信する手段とを含むようにすることも考えられる。

【００４８】本発明の他の態様としては、複数のテクセルからなるテクスチャ・イメージと、そのテクスチャ・イメージを１以上の段階で縮小したデータであり且つ１又は複数のテクセルからなる縮小テクスチャ・イメージ

とを含むテクスチャ・イメージ・データをインターリーブして重複を生じないよう各々のメモリに格納する複数のテクスチャ・メモリ・クラスタであって、ある１つのピクセルに対するテクスチャ・イメージの座標及びピクセルとテクスチャ・イメージのテクセルの縮小率から前述の１のピクセルに対するテクスチャ値を計算するのに必要なテクセルを求め、そのテクセルの色情報であるテクセルの値を複数のテクスチャ・メモリ・クラスタが出力するように要求する手段と、要求されたテクセルの値を検索し、要求したテクスチャ・メモリ・クラスタに出力する手段と、テクセルの値を収集する手段と、収集されたテクセルの値からテクスチャ値を計算する手段とをそれぞれ有する先の複数のテクスチャ・メモリ・クラスタと、複数のテスクチャ・メモリ・クラスタを接続するバスと、複数のテクスチャ・メモリ・クラスタのうちの１つにそれぞれ接続され、前述の１のピクセルに対するテクスチャ・イメージの座標及び縮小率を計算する複数のテクスチャ生成器と、複数のテクスチャ・メモリ・クラスタのうちの１つにそれぞれ接続され、前述の１のピクセルに対するテクスチャ値を用いて前述の１のピクセルに表示すべきデータを生成する複数の描画プロセッサとを有するテクスチャ・マッピングを行うコンピュータ・システムがある。これによれば、ミップマップを用いるようなテクスチャ・マッピングにおいてテクスチャ・メモリ・クラスタのメモリを有効利用することができる。

【００４９】

【作用】第１に述べた本発明の構成では、各テクスチャ生成器が１のピクセルに対するテクスチャ・イメージの座標を生成して（ラスタライズ）、それぞれに接続されたテクスチャ・メモリ・クラスタに送る。そして、各テクスチャ・メモリ・クラスタでは、収集手段がそのテクスチャ・イメージの座標から必要なテクスチャ・イメージ・データのテクセルを計算する。そして、そのテクセルを有する、自己を含むテクスチャ・メモリ・クラスタからそのテクセルのテクセル値を収集する。必要なテクセル値を収集できれば、先の１のピクセルに対するテクスチャ値を計算する。計算されたテクスチャ値は、計算を行ったテクスチャ・メモリ・クラスタに接続された描画プロセッサで、その１のピクセルに表示すべきデータを計算する時に用いられる。そして、その表示すべきデータはフレーム・バッファに入れられる。

【００５０】上述の収集手段が通知手段と検索手段と送信手段と受信手段に別れる場合には、通知手段が、必要なテクセルを自己を含む複数のテクスチャ・メモリ・クラスタに通知する。そして、検索手段がそのテクセルを有しているかメモリを検索し、有していればそのテクセルのテクセル値を出力する。出力があれば送信手段により要求元に送信し、要求元のテクスチャ・メモリ・クラスタの受信手段がそのテクセル値を受信する。

【００５１】上述のようにメモリを複数のバンクに分けており、各バンクでもテクスチャ・イメージのインターリーブが行われている場合には、その各バンクに目的とするテクセルが格納されているかを記録しておき、そのバンクからデータを引き出す必要がある。

【００５２】上述のように各メモリ・バンクに入力ＦＩＦＯと出力ＦＩＦＯを有していれば、必要なテクセル値が同一のメモリ・バンクに格納されている場合でも、一旦ＦＩＦＯに格納しておくことにより、同一メモリ・バンクへのアクセスがあっても全体のパフォーマンスの低下を抑えることができる。

【００５３】また、上述の検索手段が、アドレス計算手段と分配手段とテクセル値を得る手段とを含む場合には、通知されたテクセルのメモリ中のアドレスをそのアドレス計算手段が計算し、分配手段がそのアドレスに応じて該当メモリ・バンクにそのアドレスを分配する。そして、そのメモリ・バンクから目的のテクセル値を得ることができる。

【００５４】さらに、必要なテクセルの通信元のテクスチャ・メモリ・クラスタのＩＤを記憶するＦＩＦＯバッファを有していれば、そのテクセルの要求を受信した際に要求したテクスチャ・メモリ・クラスタをこのＦＩＦＯに入力し、目的のテクセル値を格納したメモリ・バンクからの出力をそのＦＩＦＯの記憶内容の順(要求の受信順)に送信するようにできる。

【００５５】計算されたアドレスを分配したメモリ・バンクのＩＤを記憶するＦＩＦＯバッファを有している場合には、アドレスの分配を行う際に分配したメモリ・バンクのＩＤをこのＦＩＦＯに記憶しておき、先のテクスチャ・メモリ・クラスタのＦＩＦＯと併せてどのバンク出力をどのテクスチャ・メモリ・クラスタに出力するかを対応付けする。

【００５６】収集手段の他の態様の場合には、テクスチャ・イメージの座標をそのまま、自己を含む複数のテクスチャ・メモリ・クラスタに送信し、受信したテクスチャ・メモリ・クラスタで、計算手段が必要なテクセルであって保持しているテクセルを計算する。このテクセルのテクセル値を取り出し、要求元(通信元)のテクスチャ・メモリ・クラスタに送信する。そのテクスチャ・メモリ・クラスタは、送られてきたテクセル値を受信し、フィルタにてテクスチャ値を計算することになる。

【００５７】さらに他の本発明の態様の場合には、テクスチャ生成器が、１のピクセルに対するテクスチャ・イメージの座標とそのピクセルとテクセルの縮小率を計算し、それを接続されているテクスチャ・メモリ・クラスタに送信する。そして、要求されたテクセルの値を検索し、要求したテクスチャ・メモリ・クラスタに送信する。そして必要なテクセル値を収集してテクスチャ値を計算する。計算されたテクスチャ値は、そのテクスチャ・メモリ・クラスタに接続された描画プロセッサによっ

て、上述のピクセルの表示すべきデータの計算に用いられる。その表示すべきデータはフレーム・バッファに入力され、表示装置に表示される。

【００５８】

【実施例】図１に本願発明の用いられるシステムの全体を示す。ワークステーション１は、メインＣＰＵ、メイン・メモリ、キーボード、プリンタ等の入出力装置、及びＦＤＤやＨＤＤといった記憶装置を含む。このワークステーション１のバスに接続されているのが、グラフィック・サブシステム３であり、このサブシステム３の出力が表示装置１１にて表示される。グラフィック・サブシステム３は、ジオメトリ計算部５とラスタ計算部７とテクスチャ計算部９により構成される。

【００５９】この図１の動作を説明する。ワークステーション１のＣＰＵは、表示装置１１に何等かの表示を行いたい場合には、グラフィック・サブシステム３に描画を命令する。例えば、ＣＰＵがある位置（全体座標）における、ある物体に、あるテクスチャ・イメージを張り付けて描くことを命令した場合、ジオメトリ計算部５はポリゴン分割をし、ポリゴンの頂点におけるスクリーン上の座標を求め、その座標におけるテクスチャ・イメージの座標を計算する。もし、色の指定がさらにある場合にはその色情報も計算する。この結果を用いてラスタ計算部７が、そのポリゴン内部の各ピクセルごとにスクリーン上の座標及びテクスチャ・イメージの座標（あれば色情報も）を計算する。このテクスチャ・イメージの座標とテクセルの拡大縮小率ＬＯＤ（縮小率ともいう。）からテクスチャ計算部９が、そのピクセルに対応するテクスチャの値を生成する。そして、そのテクスチャの値を用いて表示すべきデータを生成し、フレーム・バッファに格納する。格納されたデータにより表示装置１１のスクリーン上に上述の物体を描き出すようになっている。

【００６０】このグラフィック・サブシステムの詳細を図２に示す。ジオメトリ計算部５は、バス１３により複数のテクスチャ生成器／フラグメント生成器１５，１７，１９，２１，２３，２５，２７，２９（ここでは８つ）に接続されている。そして、各テクスチャ生成器／フラグメント生成器は、１のテクスチャ・メモリ・クラスタに接続されている。このテクスチャ・メモリ・クラスタ３３，３５，３７，３９，４１，４５，４７は、それぞれアドレス分配バス３１とデータ交換バス９５で接続されている。また、各テクスチャ・メモリ・クラスタは描画プロセッサ５１，５３，５５，５７，５９，６１，６５にそれぞれ接続されている。この描画プロセッサは、フラグメント生成器も直接接続されている。そして各描画プロセッサは、フレーム・バッファ６７，６９，７１，７３，７５，７７，７９，８１にそれぞれ接続されている。

【００６１】テクスチャ・メモリ・クラスタの概略はテクスチャ・メモリ・クラスタ３３にのみ示してある。他のテクスチャ・メモリ・クラスタに関しては、その内容は同一であり説明を省略する。テクスチャ生成器１５からの出力はアドレス生成器８３に接続されており、アドレス生成器８３の出力はアドレス分配バスに接続されている。また、他の出力はアドレス選択器８５に接続されている。アドレス選択器８５の出力は、アドレス分配器８７に接続され、このアドレス分配器８７は複数のメモリ・バンク８９（ここでは８つ）に接続される。メモリ・バンク８９の出力はデータ収集器９１に接続されている。このデータ収集器９１の出力はデータ交換バス９５にも接続されており、他のテクスチャ・メモリ・クラスタにデータを出力するようになっている。このデータ収集器９１の出力は、フィルタ９３にも接続されている。フィルタ９３では、このデータ収集器９１からの出力とともに、データ交換バス９５に接続されており、他のテクスチャ・メモリ・クラスタからの出力を受信するようになっている。そして、フィルタ９３は生成したテクスチャ値を担当の描画プロセッサ５１に出力するようになっている。

【００６２】以下、図２に示した装置の動作を説明する。ジオメトリ計算部５は、上述したように各ポリゴンの頂点に係るピクセルの座標とテクスチャ・イメージの座標、その点での色情報を計算し出力する。この出力はバス１３でブロードキャストされ、各テクスチャ生成器／フラグメント生成器で受信される。各テクスチャ生成器はポリゴン内部のピクセルを補間計算し、各ピクセルごとの、スクリーン上の座標、テクスチャ・イメージの座標を計算する。また、フラグメント生成器は、同様にポリゴン内部のピクセルを補間計算し、各ピクセルごとの、スクリーン上の座標、色情報を計算する。この補間計算の際には、ピクセルとテクセルの縮小率ＬＯＤも同時に計算される。但し、各テクスチャ生成器及びフラグメント生成器は担当するピクセルを予め割り当てられており、担当するピクセル以外のデータは生成されない。なお、ラスタ計算器７は図２では、テクスチャ生成器／フラグメント生成器に対応する。

【００６３】ここで、各テクスチャ・メモリ・クラスタがどのようにテクスチャ・イメージを保持しているかを簡単に説明する。なお、一般のアプリケーションでは従来技術で述べたリニアーミップマップーリニアが用いられており、この方式でテクスチャ・マッピングを行う際にパフォーマンスを最大にすることを目的とした例を述べることを予め断っておく。このリニアーミップマップーリニアでは、同時に８つのテクセル値にアクセスできなければならない。よって、必要なテクセル値が８つのテクスチャ・メモリ・クラスタに分散されて記憶されているようにする。すなわち図２の例では、例えばテクスチャ・メモリ・クラスタ３３（クラスタ０）は縮小率ＬＯＤが偶数で、ｉが偶数、ｊが偶数のテクセル、テクス

チャ・メモリ・クラスタ３５（クラスタ１）は縮小率ＬＯＤが偶数で、ｉが奇数、ｊが偶数のテクセル、テクスチャ・メモリ・クラスタ３７（クラスタ２）は縮小率ＬＯＤが偶数で、ｉが偶数、ｊが奇数のテクセル、テクスチャ・メモリ・クラスタ３９（クラスタ３）は縮小率ＬＯＤが偶数で、ｉが奇数、ｊが奇数のテクセル、テクスチャ・メモリ・クラスタ４１（クラスタ４）は縮小率ＬＯＤが奇数で、ｉが偶数、ｊが偶数のテクセル、テクスチャ・メモリ・クラスタ４３（クラスタ５）は縮小率ＬＯＤが奇数で、ｉが奇数、ｊが偶数のテクセル、テクスチャ・メモリ・クラスタ４５（クラスタ６）は縮小率ＬＯＤが奇数で、ｉが偶数、ｊが奇数のテクセル、テクスチャ・メモリ・クラスタ４７（クラスタ７）は縮小率ＬＯＤが奇数で、ｉが奇数、ｊが奇数のテクセル、とする。

【００６４】元のテクスチャ・イメージが８×８の場合の例を、図３に示す。図３（Ａ）は、ＬＯＤが０のテクスチャ・イメージを示している。この各枠内の数字は、そのテクセル値が格納されるテクスチャ・メモリ・クラスタの番号（上記のカッコ内の数字）を表す。このようにＬＯＤが０、すなわち偶数の場合には、クラスタ０からクラスタ３までに格納される。またＬＯＤが１（図３（Ｂ））の場合には、クラスタ４からクラスタ７までに格納される。図３（Ｃ）の場合も同様である。

【００６５】このようにテクスチャ・イメージが各テクスチャ・メモリ・クラスタに格納されていることを前提に以下動作の説明を続行する。テクスチャ生成器からテクスチャ・イメージの座標及び縮小率ＬＯＤを受信したテクスチャ・メモリ・クラスタは、アドレス生成器が必要なテクセルを見い出す。すなわち、計算されたＬＯＤの整数部分の値とそれに＋１した値（必ず偶数と奇数になる）と、そのレベル毎のテクセルの座標（ｉ，ｊ）を計算する。そして、それをアドレス分配バス３１に出力するとともに、接続されたアドレス選択器にも出力する。アドレス分配バス３１の出力も、それを出力したテクスチャ・メモリ・クラスタ以外のクラスタのアドレス選択器に入力される。例えば、アドレス生成器が計算したテクセルの座標を左下として、１つのＬＯＤのミップマップにつき４つのテクセル値を参照するので、各アドレス選択器は受信した座標のテクセルではなく、その４つの内の自己が保持しているテクセルが要求されていると認識し、そのテクセルの座標を計算する。他のＬＯＤの４つのテクセルについても同様である。但し、ここではアドレス選択器で計算していた処理は、要求元のアドレス生成器が１つのＬＯＤごとに４つのテクセルの座標を生成し、保持しているテクスチャ・メモリ・クラスタに送信しても同様である。

【００６６】このようにして、各アドレス選択器には、直接接続されたアドレス生成器からの要求も含めて、８つのテクセルについての要求が入力される。そして、このアドレス選択器は８つの要求を順にアドレス分配器に出力する。アドレス分配器は、テクセルの座標に対応する、メモリの物理アドレスを計算し、その物理アドレスに対応するメモリ・バンクにその物理アドレスを出力する。そして目的とするテクセル値がデータ収集器に出力され、要求元のテクスチャ・メモリ・クラスタにデータ交換バスにて送られる。もし、そのテクセル値が自己のテクスチャ・メモリ・クラスタの要求であった場合には、フィルタに直接送られる。フィルタは、上述のようにデータ交換バスにて送られてきた７つのテクセル値と、自己のデータ収集器から送られてきた１つのテクセル値で所定の計算を行い、テクスチャ値を生成する。生成されたテクスチャ値は、接続された描画プロセッサに出力される。描画プロセッサは、そのテクスチャ値が対応するスクリーン上のピクセルに対応する、フラグメント生成器により生成された色情報と、フレーム・バッファの値を参照して、そのピクセウに関する表示データを生成し、所定のフレーム・バッファに格納する。フレーム・バッファに入力されたデータは、コントローラにより読み出され、表示装置に表示される。

【００６７】以上、図２の動作を述べたが、メモリ・バンクについてさらに述べる。上で述べたように、アドレス分配器は、物理アドレスに対応するメモリ・バンクにその物理アドレスを出力する。すなわち、各テクスチャ・メモリ・クラスタ内のメモリ・バンクでも、物理アドレスでインターリーブして記憶している。ここではメモリ・バンクは８つであるから、物理アドレスの下位３桁で、インターリーブするとよい。このようにすれば、１つのテクセル値も重複することなく、テクスチャ・イメージをテクスチャ・メモリ・クラスタに格納することができ、メモリ容量の削減に有効である。例えば、第１の従来例では、通信して各プロセッサで輝度計算を行うため、各プロセッサ内のメモリに格納されるテクセルに重複が必ず生じていた。また、第２の従来例でも、１Ｍｂｙｔｅのテクスチャ・イメージを格納する際には８Ｍｂｙｔｅのメモリが必要になる。これに対し、本発明は１Ｍｂｙｔｅのテクスチャ・イメージを格納するのに１Ｍｂｙｔｅしか必要なく、大容量化しつつあるテクスチャ・イメージを大量に格納することができ、有効である。

【００６８】図２に示したテクスチャ・メモリ・クラスタの詳細を図４に示す。テクスチャ生成器の出力は、偶ＬＯＤアドレス生成器８３ａと奇ＬＯＤアドレス生成器８３ｂに接続されている。また、この２つのアドレス生成器の出力は、アドレス分配バス３１に接続される。さらに、そのテクスチャ・メモリ・クラスタの格納しているテクセルに依存して、２つのアドレス生成器のどちらかの出力が、アドレス選択器８５に接続される。このアドレス選択器８５には、バッファ１０９を介して他のテクスチャ・メモリ・クラスタからの出力が入力されるようになっている。そして、アドレス選択器８５はアドレ

ス分配器８７に接続されており、このアドレス分配器８７は複数のメモリ・バンク１０５（ここでは、８つ）に接続されている。各メモリ・バンクには、入力ＦＩＦＯバッファ１０５ａとＤＲＡＭ１０５ｃと出力ＦＩＦＯバッファ１０５ｂがある。またアドレス選択器８７は、タグＦＩＦＯバッファ１１５とあて先ＦＩＦＯバッファ１０３に接続されている。このメモリ・バンク及びタグＦＩＦＯ１１５及びあて先ＦＩＦＯ１０３の出力はデータ収集器９１に接続され、このデータ収集器９１の出力は、一方はバッファ１１３を介して他のテクスチャ・メモリ・クラスタに出力され、他方はフィルタ９３に出力されるようになっている。フィルタ９３には他のテクスチャ・メモリ・クラスタからの出力がバッファ１１１を介して入力されるようになっている。フィルタ９３の出力は、上述のように描画プロセッサに接続されている。

【００６９】以下、図４の動作を説明する。図２のアドレス生成器は、偶ＬＯＤアドレス生成器８３ａと奇ＬＯＤアドレス生成器８３ｂで構成されている。この偶ＬＯＤアドレス生成器は、偶数のＬＯＤに係るテクスチャ・イメージを格納するテクスチャ・メモリ・クラスタに接続されている。同様に、奇ＬＯＤアドレス生成器は、奇数のＬＯＤに係るテクスチャ・イメージを格納するテクスチャ・メモリ・クラスタに接続されている。当然、自己のテクスチャ・メモリ・クラスタもテクセルを格納しているから、自己の格納しているテクスチャ・イメージのＬＯＤによりマルチプレクサ１０１を操作して、アドレス選択器８５に出力する。そして、テクスチャ値を計算するのに必要な定数を定数ＦＩＦＯ１０７に格納する。必要な定数は従来技術の説明でａ，ｂ，ｄと言っていたものである。

【００７０】他のテクスチャ・メモリ・クラスタでも同様にして、例えば必要なテクセルの左下のテクセルの座標を出力するので、その出力がバッファ１０９に入力される。アドレス選択器は、バッファ１０９及びマルチプレクサ１０１の出力から１つづつ選択してきて、先ほど述べたように、左下のテクセルの座標に対応し且つ必要とする４つのテクセルの内の自己の格納しているテクセルを認識する。そしてそのテクセルの座標を計算する。但し、左下のテクセルの座標を得た段階で、必要なテクセルの格納された物理アドレスを出力するような構成にすれば、より効率的である。図２の説明では、話の流れ上前者のような動作であるとしたが、対応関係は明瞭であるから後者のようにすることは何等の困難もない。アドレス選択器８５で生成された物理アドレスは、アドレス分配器がそのアドレスの下位３桁で、分配するメモリ・バンクを認識し、そのメモリ・バンクのＩＤ（又はタグ）をタグＦＩＦＯ１１５に入力する。また、その物理アドレスに格納されたテクセル値を要求したテクスチャ・メモリ・アドレスのＩＤをあて先ＦＩＦＯ１０３に入力する。これにより、どのメモリ・バンクからの出力が、どのテクスチャ・メモリ・クラスタに出力されるかが順番を誤ることなく認識できる。物理アドレスは対応するメモリ・バンクの入力ＦＩＦＯバッファ１０５ａに入力される。このＦＩＦＯは、アドレスによりインターリーブしているために、入力される物理アドレスがある１つのメモリ・バンクに集中する場合がある。このような場合であっても、順番を誤ることなく、且つあふれることなく、出力を得るために設けたものである。出力ＦＩＦＯバッファ１０５ｂも同様の理由で設けられるものである。

【００７１】但し、入力ＦＩＦＯを設けたとしてもあふれる場合も考えられるので、あふれたかどうかもアドレス分配器は検査する。もしあふれた場合には、そのメモリ・バンクの処理が進み、入力ＦＩＦＯバッファ１０５ａに空きが生ずるまで、入力ＦＩＦＯバッファ１０５ａへの入力を停止する。そして、テクセル値は出力ＦＩＦＯバッファ１０５ｂに入力される。ここでデータ収集器９１は、タグＦＩＦＯ１１５とあて先ＦＩＦＯ１０３を１づつそれぞれ読み出す。そして、タグＦＩＦＯ１１５の示すメモリ・バンクの出力ＦＩＦＯ１０５ｂからテクセル値を読み出し、対応するバッファ１１３に出力する。または、自己の要求に基づく場合にはフィルタ９３に出力する。そして、バッファ１１３から要求元のテクスチャ・メモリ・クラスタに出力される。

【００７２】このようにしてデータ交換バスを介して得たテクセル値は、まず、バッファ１１１に格納され、そしてフィルタ９３に入力される。これにより８つの必要なテクセル値が集まったので、定数ＦＩＦＯ１０７から必要な定数を読み出し、従来技術の説明で示した計算をフィルタ９３が行う。この計算によりテクスチャ値が生成され、描画プロセッサに与えられる。

【００７３】以上述べたように、これによりメモリの有効利用を行うと共に、この例ではラスタ計算部以下を並列に行うことにより、テクスチャ・メモリ・クラスタ間の通信を行うにしても、パフォーマンスを許容範囲内の悪化にとどめることができた。しかし、これは一例であって、例えば、テクスチャ・メモリ・クラスタの数はその同時アクセスが必要なテクセルの数に合わせること、又はパフォーマンスを落とすのであれば複数回テクスチャ・メモリ・アクセスするようにして、テクスチャ・メモリ・クラスタの数を減らす等の変更を行うことができる。さらに、アドレス生成器をテクスチャ生成器に接続している例を示したが、縮小率ＬＯＤとテクスチャ・イメージの座標をそのまま複数のテクスチャ・メモリ・クラスタにブロードキャストし、アドレス選択器で計算してもよい。また、メモリ・バンクの数も、アクセスが集中しないような数に変更することは可能である。さらに、入力ＦＩＦＯバッファ及び出力ＦＩＦＯバッファの段数も、同様にメモリ・バンクの容量等に依存するので，変更可能である。

21

【００７４】さらに、あて先ＦＩＦＯ及びタグＦＩＦＯの段数もメモリ・アクセスの速度等に依存するので、変更可能である。また、８つのテクセルを同時に必要とする例を示したが、この数が増加した場合、減少した場合にも、そのインターリーブの仕方及びテクスチャ・メモリ・クラスタの数にて対応可能である。

【００７５】

【発明の効果】以上説明したように、コンピュータ・グラフィクス・インタフェイスにあるテクスチャ・マッピングの機能を効率良く実行させることができるコンピュータ・グラフィクス装置を得ることができた。

【００７６】また、必要とするメモリの容量を劇的に減少させても効率が許容範囲内の悪化で済むシステムを得ることもできた。

【００７７】具体的には、メモリ・バンクの数を８、物理アドレスはランダムである、ＤＲＡＭのアクセス時間は６０ｎｓである、各ＦＩＦＯは７．５ｎｓで動作可能、物理アドレスは７．５ｎｓごとに入力され、アドレス分配器は７，５ｎｓごとに分配することができる。データ収集器は７．５ｎｓで出力ＦＩＦＯからデータを回収できる、との条件でシミュレーションを行った結果、ＤＲＡＭの利用率は８５パーセントとなり、アクセスレートとしては約１１３ＭＨｚであった。

【図面の簡単な説明】

【図１】本発明が適用可能される装置の全体図である。

【図２】本発明が適用可能な一実施例のコンピュータ・グラフィクス装置の要部構成を示すブロック図である。

【図３】テクスチャ・メモリ・クラスタに記憶するテクセルの格納方法を示す概念図である。

【図４】テクスチャ・メモリ・クラスタ内の詳細なブロック図である。

【図５】コンピュータ・グラフィクスにおいて用いる座標系を示す線図である。

22

＊【図６】ミップマップを説明するための説明図であり、（Ａ）はレベル０のミップマップであるテクスチャイメージを示す線図、（Ｂ）はサイズが半分のレベル１のミップマップであるテクスチャ示す線図である。

【図７】従来のコンピュータグラフィクス装置の要部構成を示すブロック図である。

【図８】従来例のデータ・フォマットを示すブロック図である。

【図９】従来例のデータ・フォマットを示すブロック図である。

【図１０】従来例のデータ・フォマットを示すブロック図である。

【符号の説明】

１　ワークステーション　　３　　グラフィック・サブシステム

５　ジオメトリ計算部　　７　　ラスタ計算部

９　テクスチャ計算部　　１１　表示装置

１３　　バス

１５から２９　　テクスチャ生成器／フラグメント生成器

３１　　アドレス分配バス

３３から４７　　テクスチャ・メモリ・クラスタ

５１から６５　　描画プロセッサ

６７から８１　　フレーム・バッファ

８３　　アドレス生成器　　８５　アドレス選択器

８７　　アドレス分配器　　８９　メモリ・バンク

９１　　データ収集器　　９３　　フィルタ

９５　　データ交換バス　　１０１　　マルチプレクサ

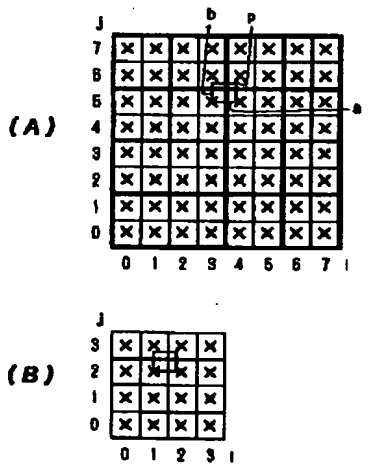
１０３　　あて先ＦＩＦＯ　　１０５　　メモリ・バンク

１０７　　定数ＦＩＦＯ　　１０９　　バッファ

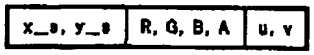
１１１　　バッファ　　１１３　　バッファ

＊　１１５　　タグＦＩＦＯ

【図６】



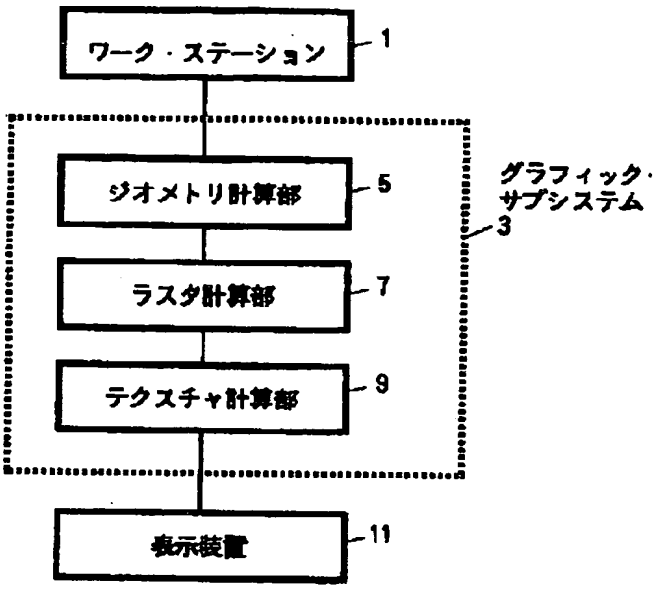
【図８】



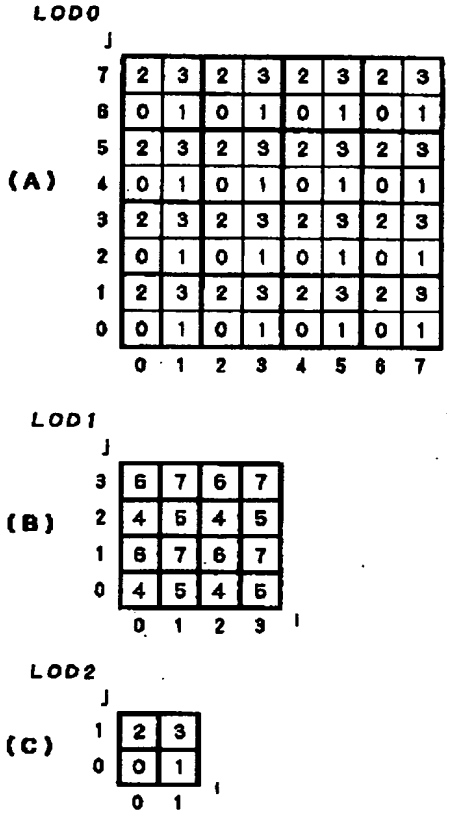
【図９】

【図１０】

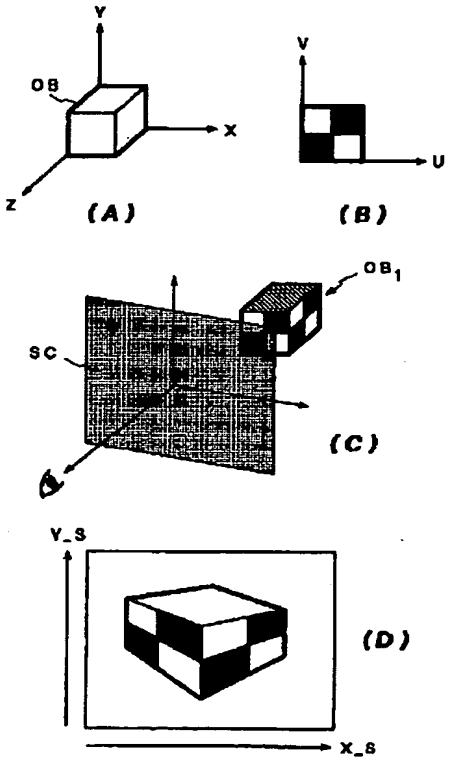
【図１】


ワーク・ステーション
1
ジオメトリ計算部
5
ラスタ計算部
7
テクスチャ計算部
9
表示装置
11
グラフィック・サブシステム
3

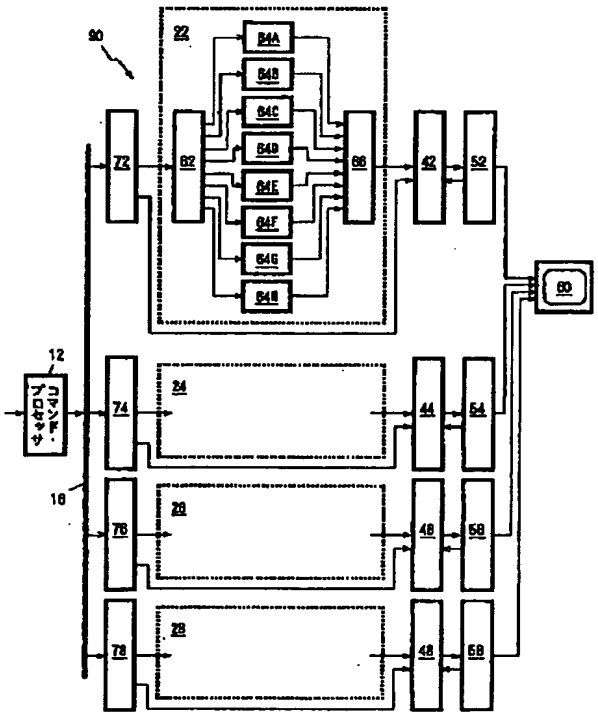
【図３】


LOD0
(A)
LOD1
(B)
LOD2
(C)

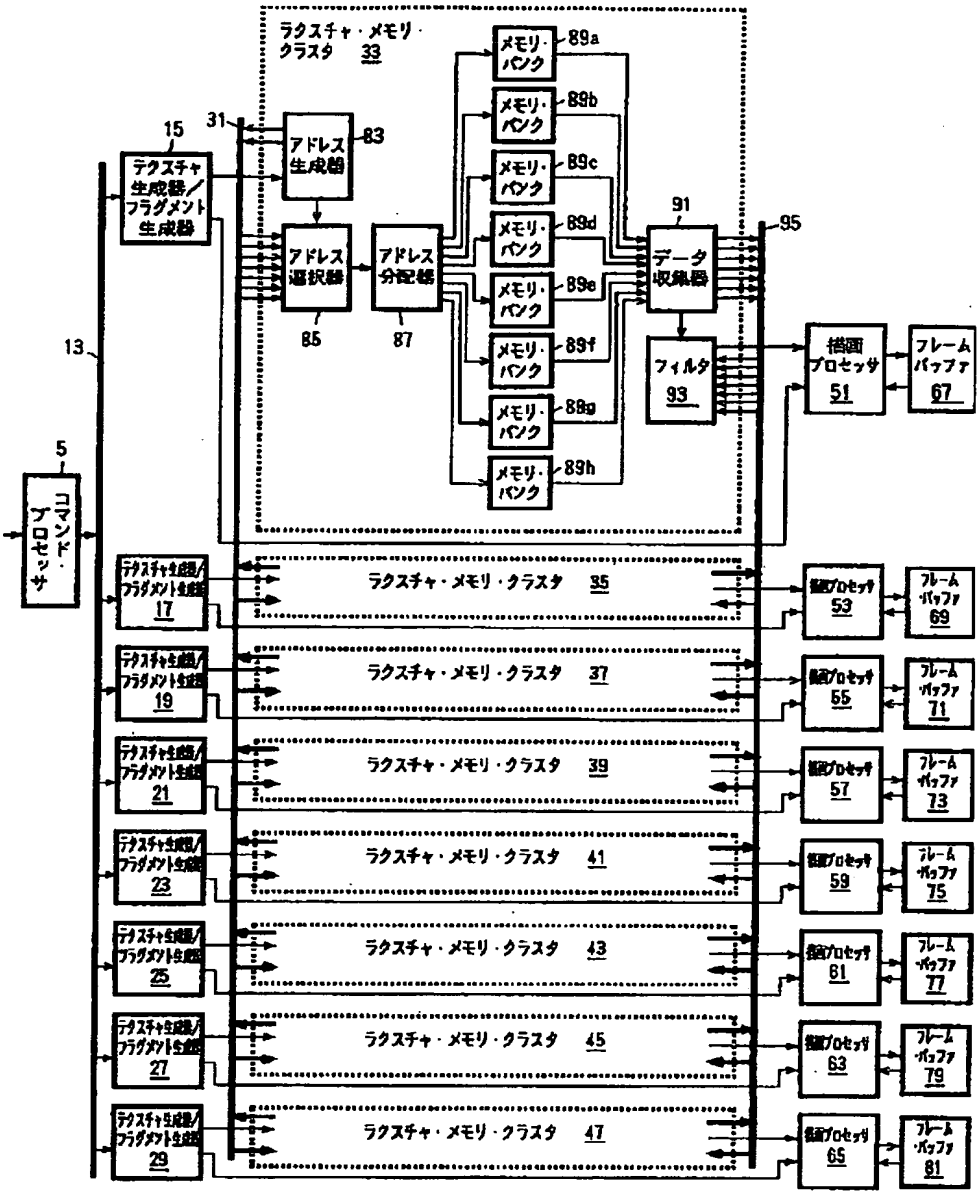
【図５】


(A)
(B)
(C)
(D)
V_S
X_S

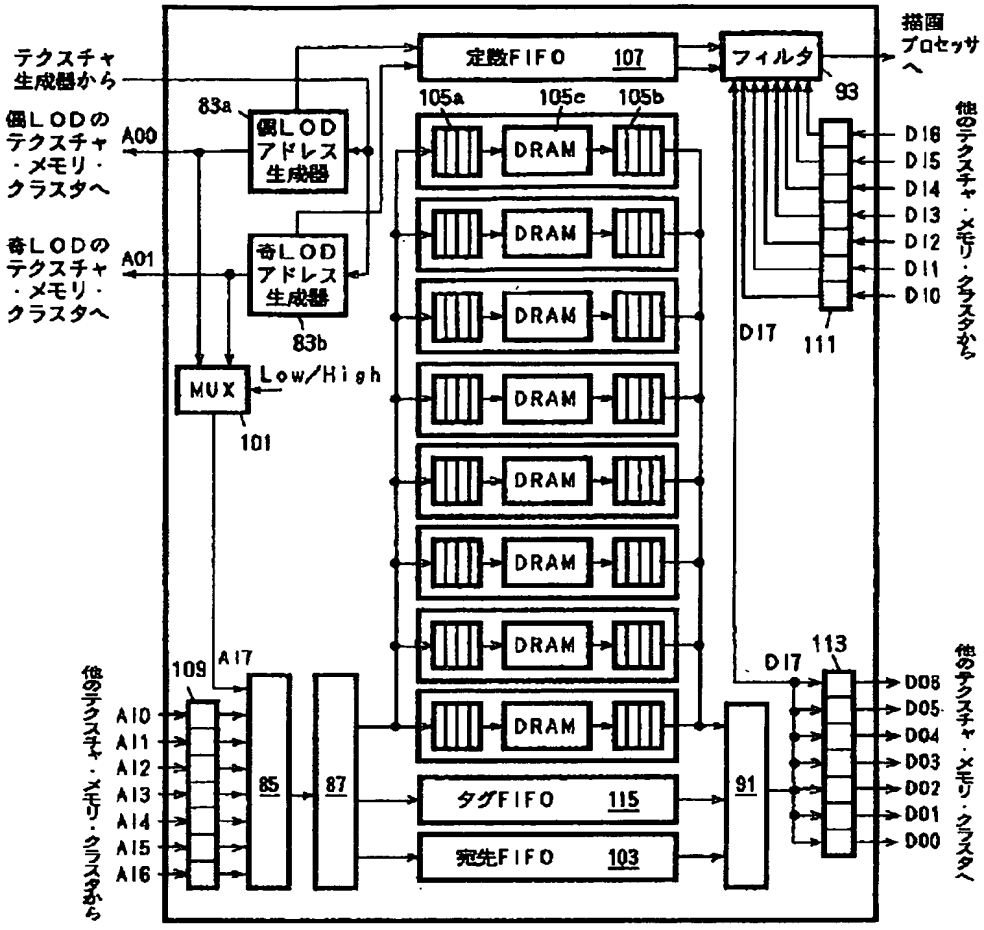
【図７】


コマンド・プロセッサ

【図２】


ラクスチャ・メモリ・クラスタ 33
メモリ・バンク 89a
メモリ・バンク 89b
メモリ・バンク 89c
メモリ・バンク 89d
メモリ・バンク 89e
メモリ・バンク 89f
メモリ・バンク 89g
メモリ・バンク 89h
15
31
アドレス生成器 83
テクスチャ生成器／フラグメント生成器
アドレス選択器 85
アドレス分配器 87
91
データ収集器
95
フィルタ 93
描画プロセッサ 51
フレームバッファ 67
13
5
コマンド・プロセッサ
ラクスチャ・メモリ・クラスタ 35
ラクスチャ・メモリ・クラスタ 37
ラクスチャ・メモリ・クラスタ 39
ラクスチャ・メモリ・クラスタ 41
ラクスチャ・メモリ・クラスタ 43
ラクスチャ・メモリ・クラスタ 45
ラクスチャ・メモリ・クラスタ 47
17
19
21
23
25
27
29
53
55
57
59
61
63
65
69
71
73
75
77
79
81

【図４】



フロントページの続き

(72)発明者　田中　伸宜
神奈川県大和市下鶴間1623番地14　日本アイ・ビー・エム株式会社　大和事業所内